Journal of Mathematics Research; Vol. 13, No. 1; February 2021
ISSN 1916-9795 E-ISSN 1916-9809
Published by Canadian Center of Science and Education

# A Feasible Approach to Determine the Optimal Relaxation Parameters in Each Iteration for the SOR Method

Chein-Shan Liu

Correspondence: Center of Excellence for Ocean Engineering, Center of Excellence for the Oceans, National Taiwan Ocean University, Keelung 202-24, Taiwan. E-mail: csliu@ntou.edu.tw

Received: December 9, 2020 Accepted: December 30, 2020 Online Published: January 7, 2021

doi:10.5539/jmr.v13n1p1 URL: https://doi.org/10.5539/jmr.v13n1p1

**Abstract**

The paper presents a dynamic and feasible approach to the successive over-relaxation (SOR) method for solving large scale linear system through iteration. Based on the maximal orthogonal projection technique, the optimal relaxation parameter is obtained by minimizing a derived merit function in terms of right-hand side vector, the coefficient matrix and the previous step values of unknown variables. At each iterative step, we can quickly determine the optimal relaxation value in a preferred interval. When the theoretical optimal value is hard to be achieved, the new method provides an alternative choice of the relaxation parameter at each iteration. Numerical examples confirm that the dynamic optimal successive over-relaxation (DOSOR) method outperforms the classical SOR method.

**Keyword:** linear equations system, successive over-relaxation (SOR) method, maximal projection, dynamic optimal relaxation parameter

## 1. Introduction

In the paper, we derive a better realization of the successive over-relaxation (SOR) method [Quarteroni, Sacco & Saleri (2000)] to find unknown variables $\mathbf{x} \in \mathbb{R}^n$ from the linear equations system:

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{n\times n}$ is a given non-singular coefficient matrix and $\mathbf{b} \in \mathbb{R}^n$ is a given right-hand side vector.

To minimize the following merit function:

$$\min_{\mathbf{x}} \left\{ f_0 = \frac{\|\mathbf{b}\|^2\|\mathbf{A}\mathbf{x}\|^2}{[\mathbf{b}\cdot(\mathbf{A}\mathbf{x})]^2} \right\}, \tag{2}$$

Liu (2013a, 2013b, 2013c, 2014a) has developed new methods to iteratively solve Equation (1). By using the Cauchy-Schwarz inequality

$$\mathbf{b}\cdot(\mathbf{A}\mathbf{x}) \leq \|\mathbf{b}\|\|\mathbf{A}\mathbf{x}\|, \tag{3}$$

it readily leads to

$$f_0 \geq 1. \tag{4}$$

In general, $f_0 > 1$ because $\mathbf{x}$ does not exactly satisfy Equation (1) during the numerical process. When $f_0 = 1$ is reached, $\mathbf{x}$ exactly satisfies Equation (1) and meanwhile the solution is obtained.

Liu (2013c, 2014b) employed a scaling invariant property of Equation (2) to derive a maximal projection solution in the Krylov space and proved that Equation (2) implies the least squares solution [Blais (2010)]:

$$\min_{\mathbf{x}}\{\|\mathbf{b}-\mathbf{A}\mathbf{x}\|^2\}. \tag{5}$$

Liu (2014c), based on Equations (2) and (5), has developed a double optimal technique for the solution of Equation (1).

The SOR is a well-known and well-developed classical iterative method, and the formulation depends on a relaxation parameter, whose optimal value needs to compute the spectral radius, defined as the absolute value of the largest eigenvalue in magnitude of the iteration matrix. Therefore, its computational burden to involve the computation of optimal relaxation parameter is great, when large scale linear problem is considered. Only for limited cases, the theoretical value of the optimal relaxation parameter is known.

Bai and Chi (2003) have chosen the optimal relaxation parameter by the asymptotically optimal successive over-relaxation method in a dynamic fashion. Wen, Meng & Wang (2013) have obtained the optimal parameter by an optimization technique based on the quasi-Chebyshev accelerated iteration method. Similarly, Meng (2014) has proposed another asymptotic approach of the optimal relaxation parameter. On the other hand, Miyatake, Sogabe & Zhang (2020) proposed an adaptive method based on the Wolfe condition to search the optimal relaxation parameter.

## 2. Successive Over-Relaxation (SOR) Method

It is known that the matrix **A** can be uniquely decomposed into

$$\mathbf{A} = \mathbf{D} - \mathbf{U} - \mathbf{L}, \tag{6}$$

where the components of these matrices are given by

$$D_{ii} = A_{ii}, \text{ if } i = j, \; D_{ij} = 0, \text{ if } i \neq j, \; i, j = 1 \ldots, n, \tag{7}$$

$$U_{ij} = -A_{ij}, \text{ if } i < j, \; U_{ij} = 0, \text{ if } i \geq j, \; i, j = 1 \ldots, n, \tag{8}$$

$$L_{ij} = -A_{ij}, \text{ if } i > j, \; L_{ij} = 0, \text{ if } i \leq j, \; i, j = 1 \ldots, n. \tag{9}$$

**D** is a diagonal matrix, **U** is a strict upper triangular matrix, while **L** is a strict lower triangular matrix.

From Equations (1) and (6) it follows an equivalent linear system:

$$\mathbf{D}\mathbf{x} - \mathbf{U}\mathbf{x} - \mathbf{L}\mathbf{x} = \mathbf{b}. \tag{10}$$

Multiplying the above equation by a nonzero constant $w$, it becomes

$$w\mathbf{D}\mathbf{x} - w\mathbf{U}\mathbf{x} - w\mathbf{L}\mathbf{x} = w\mathbf{b}. \tag{11}$$

Then adding a term **Dx** on both sides, we have

$$\mathbf{D}\mathbf{x} + w\mathbf{D}\mathbf{x} - w\mathbf{U}\mathbf{x} - w\mathbf{L}\mathbf{x} = w\mathbf{b} + \mathbf{D}\mathbf{x}. \tag{12}$$

By using an iterative method to solve the linear system, we suppose that the value $\mathbf{x}_k$ at the $k$th-step is known. Now, we remove $w\mathbf{D}\mathbf{x}$ and $-w\mathbf{U}\mathbf{x}$ to the right-side and take the value of $\mathbf{x}$ in the the right-side to be $\mathbf{x}_k$, while in the left-side we take the value of $\mathbf{x}$ to be $\mathbf{x}_{k+1}$, and then the SOR iterative method is created as follows [Hadjidimos (2000)]:

$$(\mathbf{D} - w\mathbf{L})\mathbf{x}_{k+1} = w\mathbf{b} + (1 - w)\mathbf{D}\mathbf{x}_k + w\mathbf{U}\mathbf{x}_k, \tag{13}$$

of which $0 < w < 2$ is a relaxation parameter to guarantee the convergence of iteration. If $D_{ii} \neq 0, \; i = 1, \ldots, n$, from Equation (13) it is easy to find $\mathbf{x}_{k+1}$ by using the forward substitution method, because $\mathbf{D} - w\mathbf{L}$ is a lower triangular matrix.

When **A** is a positive definite matrix, the following $w$ is the optimal one [Quarteroni, Sacco & Saleri (2000)]:

$$w_{\text{opt}} = \frac{2}{1 + \sqrt{1 - \rho^2(\mathbf{I}_n - \mathbf{D}^{-1}\mathbf{A})}}, \tag{14}$$

where $\rho$ is the spectral radius of $\mathbf{I}_n - \mathbf{D}^{-1}\mathbf{A}$.

## 3. Maximizing Orthogonal Projection

In the classical SOR method, one needs to compute the spectral radius of $\mathbf{I}_n - \mathbf{D}^{-1}\mathbf{A}$ as shown in Equation (14), which requires a huge amount of calculation work. In general, the approximate manner to try $w$ and to observe the convergence effect is adopted [Milewski & Orkisz (2014); Yang & Gobbert (2009)].

Let

$$\mathbf{y} := \mathbf{A}\mathbf{x}. \tag{15}$$

The minimization in Equation (2) is equivalent to maximize

$$\max_{\mathbf{y}} \left\{ \frac{(\mathbf{b} \cdot \mathbf{y})^2}{\|\mathbf{b}\|^2 \|\mathbf{y}\|^2} \right\}, \tag{16}$$

which is the orthogonal projection of $\mathbf{b}/\|\mathbf{b}\|$ onto the $\mathbf{y}$ direction.

In order to obtain the optimal value of $w$ in Equation (13), we temporarily take $\mathbf{x}_{k+1}$ in the left-hand side to be $\mathbf{x}_k$, and then at each iterative step, we have a linear system:

$$(\mathbf{D} - w_k\mathbf{L})\mathbf{x}_k = \mathbf{b}_1^k + w_k\mathbf{b}_2^k, \tag{17}$$

where

$$\mathbf{b}_1^k = \mathbf{D}\mathbf{x}_k,\ \mathbf{b}_2^k = \mathbf{b} - \mathbf{D}\mathbf{x}_k + \mathbf{U}\mathbf{x}_k. \tag{18}$$

Let

$$\mathbf{a}_1^k := \mathbf{D}\mathbf{x}_k,\ \mathbf{a}_2^k := -\mathbf{L}\mathbf{x}_k, \tag{19}$$

and insert them into Equation (2), we can derive

$$f_0 = \frac{f_1(w_k)f_2(w_k)}{f_3^2(w_k)}, \tag{20}$$

$$f_1(w_k) := \|\mathbf{a}_1^k + w_k\mathbf{a}_2^k\|^2 = \|\mathbf{a}_1^k\|^2 + 2w_k\mathbf{a}_1^k \cdot \mathbf{a}_2^k + w_k^2\|\mathbf{a}_2^k\|^2, \tag{21}$$

$$f_2(w_k) := \|\mathbf{b}_1^k + w_k\mathbf{b}_2^k\|^2 = \|\mathbf{b}_1^k\|^2 + 2w_k\mathbf{b}_1^k \cdot \mathbf{b}_2^k + w_k^2\|\mathbf{b}_2^k\|^2, \tag{22}$$

$$\begin{aligned} f_3(w_k) &:= (\mathbf{a}_1^k + w_k\mathbf{a}_2^k) \cdot (\mathbf{b}_1^k + w_k\mathbf{b}_2^k) \\ &= \mathbf{a}_1^k \cdot \mathbf{b}_1^k + w_k(\mathbf{a}_1^k \cdot \mathbf{b}_2^k + \mathbf{a}_2^k \cdot \mathbf{b}_1^k) + w_k^2\mathbf{a}_2^k \cdot \mathbf{b}_2^k. \end{aligned} \tag{23}$$

At each iterative step, we can search the optimal value of $w_k$ to minimize $f_0$:

$$\min_{w_k \in (a,b)} \left\{ f_0 = \frac{f_1(w_k)f_2(w_k)}{f_3^2(w_k)} \right\} \tag{24}$$

in a given interval $(a, b) \subset (0, 2)$, by taking $w_k^j = a + j(b - a)/N_w,\ j = 1, \ldots, N_w - 1$.

We can quickly find the optimal value of $w_k$ in a preferred interval $(a, b)$ for the convergence of the SOR. In this study, we use this process to pick up $w_k$ and the resulting iterative algorithm is very time saving and is named a dynamic optimal SOR (DOSOR) method. Instead of using the constant value of $w$ in the SOR method, the present method in Equation (24) provides a dynamic value of $w_k$ at each iteration.

When

$$\|\mathbf{x}_{k+1} - \mathbf{x}_k\| < \varepsilon_1, \tag{25}$$

the iteration process is terminated for satisfying a given convergence criterion. As mentioned below Eq. (4), when $f_0 \longrightarrow 1$, the solution is obtained. In this regard, we can also take the following as a convergence criterion:

$$f_0 - 1 < \varepsilon_2, \tag{26}$$

where $\varepsilon_1$ and $\varepsilon_2$ are small positive numbers.

## 4. Numerical Verification

In order to compare the performance of the DOSOR to the SOR, we test some linear problems.

### *4.1 Example 1*

Equation (1) is solved with

$$\mathbf{A} = \begin{bmatrix} 4 & -1 & 0 & 0 & 0 & 0 \\ 2 & 2 & 1.5 & 0 & 0 & 0 \\ 0 & 1 & 3 & -1 & 0 & 0 \\ 0 & 0 & 1.5 & 2 & 2 & 0 \\ 0 & 0 & 0 & 1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 2 & 2 \end{bmatrix},\ \mathbf{b} = \begin{bmatrix} 3 \\ 5.5 \\ 3 \\ 5.5 \\ 4 \\ 4 \end{bmatrix},\ \mathbf{x} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}. \tag{27}$$

From Equation (14) $w_{\text{opt}}$ is found to be

$$w_{\text{opt}} = 1.016288735, \tag{28}$$

and the SOR is convergence with 28 iterations as shown in Figure 1(a) with the following initial guess:

$$\mathbf{x}_0 = \begin{bmatrix} 10 \\ 30 \\ -20 \\ -40 \\ -8 \\ 9 \end{bmatrix}. \tag{29}$$

The maximum error (ME) is $5.71 \times 10^{-11}$ and the root-mean-square-error (RMSE) is $2.98 \times 10^{-11}$. Because $w_{\rm opt}$ is given with one calculation, the CPU time of the SOR is very short with 0.2 s.

By using the DOSOR, we take $a = 0.9$, $b = 1$ and $N_w = 10$, and through 26 iterations it converges under $\varepsilon_1 = 10^{-10}$ as shown in Figure 1(b). Because $N_w$ is small with $N_w = 10$ calculations to determine $w_k$ at each iteration, the CPU time of the DOSOR is still very short with 0.25 s. In Figure 1(b), the values of $f_0$ are plotted, which fast tends to the minimal value $f_0 = 1$, and in Figure 1(c), the values of $w$ are plotted, which are varying between $[0.9, 1]$, where the peak value is close to the optimal value given in Equation (28). The ME obtained by the DOSOR is $2.41 \times 10^{-11}$ and the RMSE is $1.27 \times 10^{-11}$, which are better than that obtained by the SOR.

If we raise $N_w$ to $N_w = 100$, the CPU time of the DOSOR is slightly increased to 0.3 s, and the ME and the RMSE become $4.85 \times 10^{-11}$ and $2.35 \times 10^{-11}$, respectively. Larger $N_w$ would increase the CPU time but not necessarily serves better accuracy. If we change the right-hand side to

$$\mathbf{b} = \mathbf{Ax} = \begin{bmatrix} 4 & -1 & 0 & 0 & 0 & 0 \\ 2 & 2 & 1.5 & 0 & 0 & 0 \\ 0 & 1 & 3 & -1 & 0 & 0 \\ 0 & 0 & 1.5 & 2 & 2 & 0 \\ 0 & 0 & 0 & 1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 2 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix},$$

the number of iterations, the CPU time and the accuracy of the DOSOR are not affected apparently. Of course, for different coefficient matrix $\mathbf{A}$, the number of iterations, the CPU time and the accuracy of the DOSOR are different as to be shown below.

### *4.2 Example 2*

In this example, we apply the DOSOR to solve the following boundary value problem:

$$u''(x) = f(x), \ u(0) = 0, \ \ u(1) = 0. \tag{30}$$

The exact solution is supposed to be

$$u(x) = \sin \pi x, \tag{31}$$

and $f(x) = -\pi^2 \sin \pi x$.

The finite difference discretization of Equation (30) is

$$\begin{aligned} &\frac{1}{(\Delta x)^2}(u_{i+1} - 2u_i + u_{i-1}) - f(x_i) = 0, \ i = 1, \ldots, n, \\ &u_0 = 0, \ \ u_{n+1} = 0, \end{aligned} \tag{32}$$

where $\Delta x = 1/(n+1)$ and $x_i = i\Delta x = i/(n+1), \ i = 1, \ldots, n$.

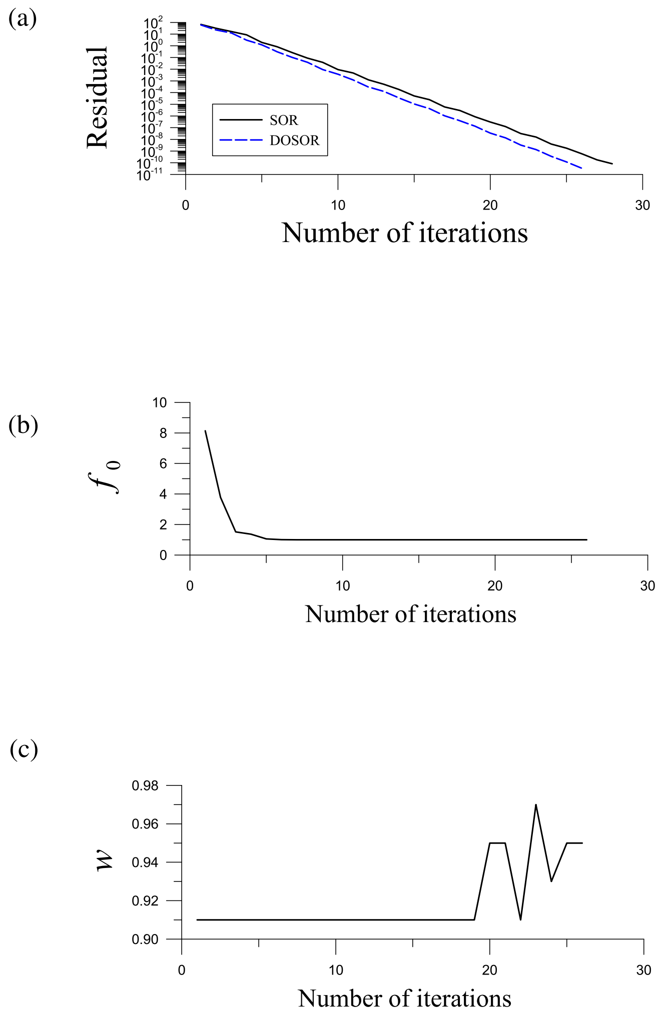


Figure 1. For example 1, (a) comparing the convergence residuals obtained by the SOR with optimal parameter and the DOSOR, (b) the minimal values of merit function and (c) optimal values of relaxation parameter

By using the DOSOR, we take $n = 99$, $a = 1.85$, $b = 1.95$ and $N_w = 10$, and through 467 iterations it converges under $\varepsilon_1 = 10^{-5}$ as shown in Figure 2(a). The CPU time of the DOSOR is very short with 0.6 s. In Figure 2(b), the values of $f_0$ are plotted, which tends to the minimal value $f_0 = 1$ very fast, and in Figure 2(c), the values of $w$ are plotted, which are varying between $[1.86, 1.94]$, where the peak value is near to the optimal value given below. The pattern is almost periodic, which is due to the periodic solution in Eq. (31) being searched. The ME obtained by the DOSOR is $2.13 \times 10^{-5}$ and the RMSE is $1.4 \times 10^{-5}$.

If we raise the value of $N_w = 100$, the CPU time increases to 2.3 s, but the accuracy is not affected with ME=$2.71 \times 10^{-5}$ and RMSE=$1.48 \times 10^{-5}$. Larger $N_w$ would increase the CPU time but not necessarily offers better accuracy. So we prefer to use $N_w = 10$ for saving CPU time. If we adopt $u(x) = x^3 - x^2$ as being another solution, the new right-hand side is obtained to be $f(x) = 6x - 2$. By using the DOSOR, we take $n = 99$, $a = 1.85$, $b = 1.95$ and $N_w = 10$, and through 660 iterations it converges under $\varepsilon_1 = 10^{-5}$. For this case the pattern of $w$ vs. number of iterations is not periodic as the one in Figure 2(c) and only with a single value $w = 1.86$. The CPU time of the DOSOR is very short with 0.8 s. ME=$1.04 \times 10^{-4}$ and RMSE=$7.35 \times 10^{-5}$ are obtained by the DOSOR.

On the other hand, we have [Demmel (1997); Watkins (2002); Yang & Gobbert (2009)]

$$w_{\text{opt}} = \frac{2}{1 + \sin(\pi\Delta x)} = 1.939091659, \tag{33}$$

which is inserted into the SOR to find the solution. It converges through 202 iterations, faster than the DOSOR; however, the maximum error obtained by the SOR with the above $w_{\text{opt}}$ is $9.27 \times 10^{-5}$ and the RMSE is $6.74 \times 10^{-5}$, which are less accurate than that obtained by the DOSOR with ME=$2.11\times10^{-5}$ and RMSE=$1.4\times10^{-5}$. Because $w_{\text{opt}}$ is given by Eq. (33) with one calculation, the CPU time of the SOR is very short with 0.5 s. The CPU times of the SOR and the DOSOR are competitive if $N_w = 10$ is taken in the DOSOR. The accuracy of the presented method is enhanced for the use of the merit function (24).

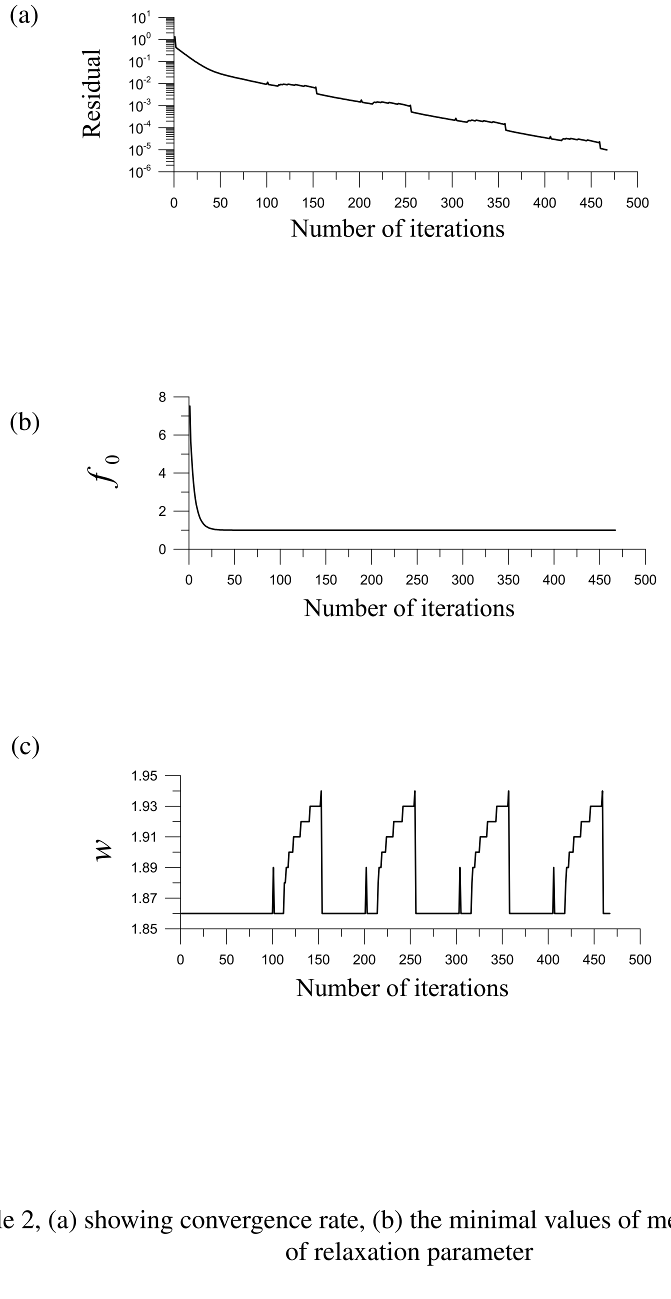


Figure 2. For example 2, (a) showing convergence rate, (b) the minimal values of merit function and (c) optimal values of relaxation parameter

### *4.3 Example 3*

We consider the Hilbert matrix:

$$A_{ij} = \frac{1}{i + j - 1} \tag{34}$$

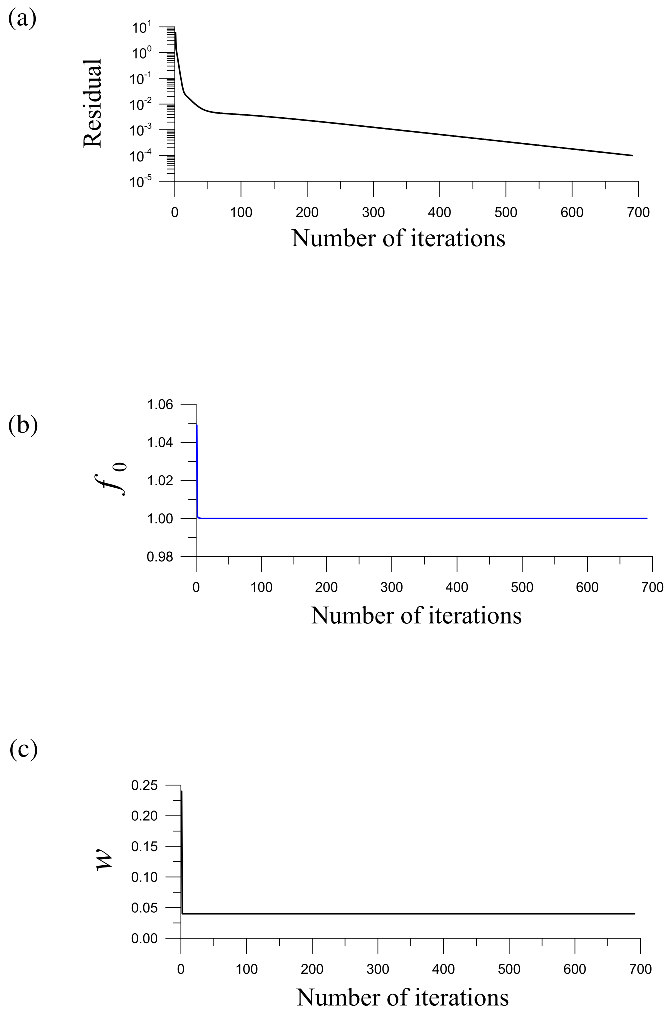


Figure 3. For example 3, (a) showing convergence rate, (b) the minimal values of merit function and (c) optimal values of relaxation parameter

in Equation (1), which is a notorious example of highly ill-conditioned matrices. Also, $x_j = 1,\ j = 1, \ldots, n$ is an exact solution, leading to

$$b_i = \sum_{j=1}^{n} \frac{1}{i+j-1}. \tag{35}$$

By using the DOSOR, we take $n = 99$, $a = 0$, $b = 2$ and $N_w = 50$, and through 691 iterations it converges under $\varepsilon_1 = 10^{-4}$ as shown in Figure 3(a). The CPU time of the DOSOR is quite short with 1.7 s. In Figure 3(b), the values of $f_0$ are plotted, which tends to the minimal value $f_0 = 1$ very fast, and in Figure 3(c) the values of $w$ are plotted, which are varying between $[0.04, 0.24]$ and tend to 0.04 very fast. The ME obtained by the DOSOR is $1.5 \times 10^{-2}$ and the RMSE is $4.56 \times 10^{-3}$. With an ad hoc value of $w = 1.5$, the number of iterations becomes 3297 and the ME and RMSE raise to $7.3 \times 10^{-2}$ and $1.36 \times 10^{-2}$, respectively.

### *4.4 Example 4*

In this example, we apply the DOSOR to solve the following 2D boundary value problem of the Poisson elliptic equation:

$$\Delta u(x, y) = p(x, y). \tag{36}$$

While

$$u(x,y) = (x^2 - x)(y^2 - y) \tag{37}$$

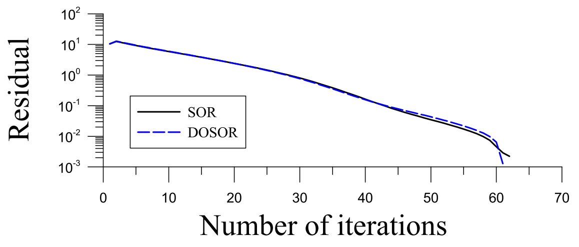
is an exact solution, $p(x,y) = 2(x^2 - x) + 2(y^2 - y)$ is derived.

As shown by Bai & Chi (2003), we take $n_1 = n_2 = 31$, $\Delta x = \Delta y = 1/32$ and $\varepsilon_1 = \Delta x^2 \|\mathbf{r}_0\|/5$, where $\mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{x}_k$ is the residual vector. When

$$w_{\text{opt}} = \frac{2}{1 + \sin(\pi \Delta x)} = 1.81072744298 \tag{38}$$

is inserted into the SOR, it converges through 62 iterations as shown in Figure 4(a). The CPU time of the SOR is very short with 0.25 s. By using the DOSOR, we take $a = 1.8$, $b = 2$ and $N_w = 10$, and through 61 iterations it converges as shown in Figure 4(a) with the CPU time being 6.86 s. The ME obtained by the SOR is $3.66\times10^{-6}$ and RMSE=$1.42\times10^{-6}$, which is less accurate than that obtained by the DOSOR with ME=$1.52 \times 10^{-6}$ and RMSE=$5.59 \times 10^{-7}$. The accuracy of the presented method is enhanced owing to the use of the merit function (24).

In the use of Equation (26), the DOSOR converges with 69 iterations under the convergence criterion $\varepsilon_2 = 10^{-12}$. However, it is deserved because the accuracy is raised to ME=$4.4 \times 10^{-7}$ and RMSE=$1.57 \times 10^{-7}$.

(a)


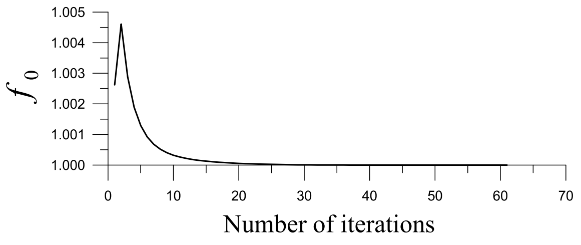
(b)


Figure 4. For example 4, (a) comparing the convergence residuals obtained by the SOR with optimal parameter and the DOSOR, and (b) the minimal values of merit function

## 5. Conclusions

There are two factors to evaluate the performance of a newly developed iterative scheme: accuracy and convergence speed. In this paper, we proposed a dynamic optimal SOR method based on the maximal orthogonal projection technique, which is equivalent to the minimization in Equation (24). The features of the proposed method are summarized as follows: (a) The merit function is in terms of the coefficient matrix, right-side vector and the value of unknown vector at the previous step. (b) Searching the minimization in a preferred interval through a few operations is easily performed. (c) The accuracy is better than the classical SOR with optimal relaxation parameter about two to four times in the maximum error as well as in the root-mean-square-error. (d) The convergence speeds are competitive of both methods. The CPU time of the DOSOR with low number of $N_w$ is slightly increased than the SOR. (e) When the theoretical value of the optimal relaxation parameter is in general not available, the new method provided an alternative and feasible choice of the relaxation parameter at each iteration.

## References

Bai, Z. Z., & Chi, X. B. (2003) Asymptotically optimal successive overrelaxation method for systems of linear equations. *J. Comput. Math., 21*, 603-612.

Blais, J. A. (2010). Least squares for practitioners. *Mathematical Problems in Engineering, 2010.* https://dx.doi.org/10.1155/508092

Demmel, J. W. (1997). Applied Numerical Linear Algebra, SIAM. https://doi.org/10.1137/1.9781611971446

Hadjidimos, A. (2000). Successive overrelaxation (SOR) and related methods. *Journal of Computational and Applied Mathematics, 123*(1-2), 177-199. https://doi.org/10.1016/S0377-0427(00)00403-9

Liu, C. S. (2013a). An optimal tri-vector iterative algorithm for solving ill-posed linear inverse problems. *Inv. Prob. Sci. Eng., 21*, 650-681. https://dx.doi.org/ 10.1080/17415977.2012.717077

Liu, C. S. (2013b). A dynamical Tikhonov regularization for solving ill-posed linear algebraic systems. *Acta Appl. Math., 123*, 285-307. https://dx.doi.org/10.1007/s10440-012-9766-3

Liu, C. S. (2013c). Discussing a more fundamental concept than the minimal residual method for solving linear system in a Krylov subspace. *J. Math. Research, 5*(4), 58-70. https://doi.org/10.5539/jmr.v5n4p58

Liu, C. S. (2014a). A globally optimal tri-vector method to solve an ill-posed linear system. *J. Comp. Appl. Math., 260*, 18-35. https://dx.doi.org/10.1016/ j.cam.2013.09.017

Liu, C. S. (2014b). A maximal projection solution of ill-posed linear system in a column subspace, better than the least squares solution. *Comput. Math. Appl., 67*, 1998-2014. https://doi.org/10.1016/j.camwa.2014.04.011

Liu, C. S. (2014c). A doubly optimized solution of linear equations system expressed in an affine Krylov subspace. *J. Comp. Appl. Math., 260*, 375-394. https://dx.doi.org/10.1016/j.cam.2013.10.013

Meng, G. Y. (2014). A practical asymptotical optimal SOR method. *Appl. Math. Comput., 242*, 707-715. https://doi.org/10.1016/j.amc.2014.06.034

Milewski, S., & Orkisz, J. (2014). In search of optimal acceleration approach to iterative solution methods of simultaneous algebraic equations. *Comput. Math. Appl., 68*, 101-117. https://doi.org/10.1016/j.camwa.2014.05.010

Miyatake, Y., Sogabe, T., & Zhang, S. L. (2020) Adaptive SOR methods based on the Wolfe conditions. *Numer. Algo., 84*, 117-132. https://doi.org/10.1007/s11075-019-00748-0

Quarteroni, A., Sacco, R., & Saleri, F. (2000). *Numerical Mathematics.* Springer Science, New York.

Trefethen, L. N., & Bau, III, D. (1997). *Numerical Linear Algebra.* SIAM, Pennsylvania. https://doi.org/10.1137/1.9780898719574

Wen, R. P., Meng, G. Y., & Wang, C. L. (2013) Quasi-Chebyshev accelerated iteration methods based on optimization for linear systems. *Comput. Math. Appl., 66*, 934-942. https://doi.org/10.1016/j.camwa.2013.06.016

Watkins, D. S. (2002). *Fundamentals of Matrix Computations* (2nd ed.). Wiley, New York. https://doi.org/10.1002/0471249718

Yang, S., & Gobbert, M. K. (2009) The optimal relaxation parameter for the SOR method applied to the Poisson equation in any space dimensions. *Appl. Math. Lett., 22*, 325-331. https://doi.org/10.1016/j.aml.2008.03.028

## Copyrights